Nomor Research GmbH

CTA WAVE Conformance Unit Testing

Nomor Research GmbH
Munich, Germany
info@nomor.de

31 October 2018

Nomor Research GmbH
Brecherspitzstraße 8
81541 München

Table of Contents

Table of Figures

1 Introduction

This document introduces Unit testing for the conformance checks implemented for the CTA WAVE content specification. The necessary installation steps are provided in Section 2. Section 3 discusses the process of unit testing used. Section 4 gives examples of unit tests for media profile and selection set conformance checks.

2 Installation

The PHPUnit package is used for unit testing and the installation is followed from the website https://phpunit.de/getting-started/phpunit-6.html

Here PHPUnit version 6 is shown, user can install PHPUnit versions supporting their installed PHP versions.

PHPUnit can be installed using PHP Archive or Composer as mentioned in the link above.

```
wget -O phpunit https://phar.phpunit.de/phpunit-6.phar

chmod +x phpunit
```

or

```
composer require --dev phpunit/phpunit ^6
```

The installation is done in ‘UnitTests’ folder inside *CTAWAVE* submodule repository of *IntegratedConformance*. [2]

3 How to write unit tests

The following example demonstrates the process of unit testing.
Note: the example is not an actual conformance check, it is only for demonstration purpose.

If the source code (Let’s call it *source.php*) has a unit/function ‘*checkProfiles’* as below.

source.php

```
function checkProfiles($x, $y)
{
    return ($x->getAttribute(‘profiles’)== $y->getAttribute(‘profiles’));
}
```

The function takes two arguments *$x* and *$y* and returns true if attribute ‘profiles’ of both arguments are equal.

To test this functionality, the unit test will be as follows, using the PHPUnit framework.
Let’s call it *sourceTest.php*

sourceTest.php

```
declare(strict_types=1);
require_once 'source.php';

use PHPUnit\Framework\TestCase;

final class sourceTest extends TestCase
{
   public function testSameProfile()
   {
              $x_string=”<AdaptationSet profiles=’cmfc’></AdaptationSet>”;
              $x=simplexml_load_string($x_string);

              $y_string=”<AdaptationSet profiles=’cmfc’></AdaptationSet>”;
              $y= simplexml_load_string($y_string);

              $this->assertTrue(checkProfiles($x, $y));
   }
}
```

The test function ‘*testSameProfile’* is a success when ‘*checkProfiles*’ function outputs true (last line).

To run this test, the following command is run from the same folder of this test script.

```
./vendor/bin/phpunit   sourceTest.php
```

The result of the unit testing is as follows in case the test is passed.

```
OK(1 test , 1 assertions)
```

In case of failure, the location of the error is displayed.

4 CTA WAVE Unit tests

In this section, the Unit tests written for CTA Wave conformance checks are discussed.

4.1 Example 1

To identify/check the conforming media profile of a track using the parameters present in the mp4 file boxes. The parameters (defined by *$xmlParamsTobeTested*) are assumed to be of AVC HD profile.

```
public function testAVCMediaProfile()
  {

     $rep_count=0;
     $adapt_count=0;

     //Choose parameters conforming to HD profile of AVC.
     $xmlParamsTobeTested=array("codec" => "AVC", "profile" => "high", "level" => "4.0",
                   "height" => "1080" , "width"=> "1920" , "framerate" =>  "60","color_primaries" => "0x1",
                   "transfer_char" => "0x1", "matrix_coeff" => "0x1", "tier" => "", "brand"=>" "  );

     $this->assertSame("AVC_HD",          checkAndGetConformingVideoProfile($xmlParamsTobeTested,          $rep_count,
$adapt_count)[0]);
}
```

The function '*checkAndGetConformingVideoProfile*' of the source code is tested here (The conformance to be tested is implemented according to the section 4.2.1 of [1]). It is expected to return 'AVC_HD'. The *$rep_count* and *$adapt_count* is required to print the error statements in case of conformance error in the source function. Hence, they are provided with dummy values in unit testing above.

4.2 Example 2

To test the Selection Set conformance checks, implemented as per the section 4.1 of [1]. The conformance check specifies that the WAVE content SHALL include one or more CMAF Track(s) and Switching Set(s) conforming to at least one WAVE approved CMAF Media Profile for each Selection Set. The following unit test is written to test one part of this conformance check, i.e., when there is no track conforming to WAVE approved profile in the Selection Set.

```
public function testNoWAVETracksInSelSet()
```

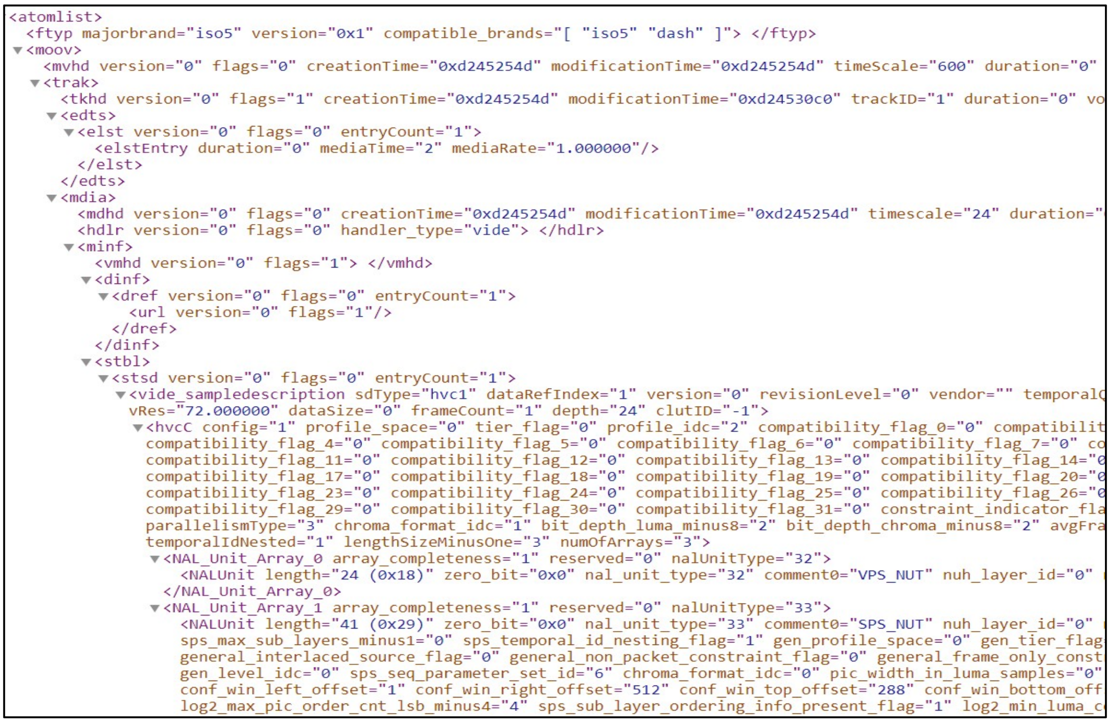
```
  {
    $adapts_count=1;
    $session_dir="Selection_set_examples/SelSetVideoNoWaveTracks";
    $adaptation_set_template='Adapt$AS$';
    $outfile=fopen("out.txt","w");

    $this->assertContains("no Tracks found conforming to WAVE",
CTACheckSelectionSet($adapts_count,$session_dir,$adaptation_set_template,$outfile));
  }
```

The source code function ‘*CTACheckSelectionSet’* is expected to return error message containing the text – ‘no Tracks found conforming to WAVE’. Some of the required inputs to the source function are provided as required.

The *$session_dir* points to a video selection set with two tracks (box level info) and both of them are non-conforming to WAVE profile. Here the tracks refer to the atom level information stored as an xml file. The xml file will be generated by the ‘ISOSegmentValidator’ submodule in the ‘IntegratedConformance’, for every mp4 track file. For unit testing purpose, they are pre-generated and kept in a folder pointed by *$session_dir*.

A snapshot of sample xml file of a video track is shown in Figure 1.



```
<atomlist>
  <ftyp majorbrand="iso5" version="0x1" compatible_brands="[ "iso5" "dash" ]"> </ftyp>
 ▼<moov>
    <mvhd version="0" flags="0" creationTime="0xd245254d" modificationTime="0xd245254d" timeScale="600" duration="0"
  ▼<trak>
      <tkhd version="0" flags="1" creationTime="0xd245254d" modificationTime="0xd24530c0" trackID="1" duration="0" vo
    ▼<edts>
      ▼<elst version="0" flags="0" entryCount="1">
          <elstEntry duration="0" mediaTime="2" mediaRate="1.000000"/>
        </elst>
      </edts>
    ▼<mdia>
        <mdhd version="0" flags="0" creationTime="0xd245254d" modificationTime="0xd245254d" timescale="24" duration="
        <hdlr version="0" flags="0" handler_type="vide"> </hdlr>
      ▼<minf>
          <vmhd version="0" flags="1"> </vmhd>
        ▼<dinf>
          ▼<dref version="0" flags="0" entryCount="1">
              <url version="0" flags="1"/>
            </dref>
          </dinf>
        ▼<stbl>
          ▼<stsd version="0" flags="0" entryCount="1">
            ▼<vide_sampledescription sdType="hvc1" dataRefIndex="1" version="0" revisionLevel="0" vendor="" temporalQ
              vRes="72.000000" dataSize="0" frameCount="1" depth="24" clutID="-1">
              ▼<hvcC config="1" profile_space="0" tier_flag="0" profile_idc="2" compatibility_flag_0="0" compatibilit
                compatibility_flag_4="0" compatibility_flag_5="0" compatibility_flag_6="0" compatibility_flag_7="0" co
                compatibility_flag_11="0" compatibility_flag_12="0" compatibility_flag_13="0" compatibility_flag_14="0
                compatibility_flag_17="0" compatibility_flag_18="0" compatibility_flag_19="0" compatibility_flag_20="0
                compatibility_flag_23="0" compatibility_flag_24="0" compatibility_flag_25="0" compatibility_flag_26="0
                compatibility_flag_29="0" compatibility_flag_30="0" compatibility_flag_31="0" constraint_indicator_fla
                parallelismType="3" chroma_format_idc="1" bit_depth_luma_minus8="2" bit_depth_chroma_minus8="2" avgFra
                temporalIdNested="1" lengthSizeMinusOne="3" numOfArrays="3">
                ▼<NAL_Unit_Array_0 array_completeness="1" reserved="0" nalUnitType="32">
                    <NALUnit length="24 (0x18)" zero_bit="0x0" nal_unit_type="32" comment0="VPS_NUT" nuh_layer_id="0"
                  </NAL_Unit_Array_0>
                ▼<NAL_Unit_Array_1 array_completeness="1" reserved="0" nalUnitType="33">
                    <NALUnit length="41 (0x29)" zero_bit="0x0" nal_unit_type="33" comment0="SPS_NUT" nuh_layer_id="0"
                    sps_max_sub_layers_minus1="0" sps_temporal_id_nesting_flag="1" gen_profile_space="0" gen_tier_flag
                    general_interlaced_source_flag="0" general_non_packet_constraint_flag="0" general_frame_only_const
                    gen_level_idc="0" sps_seq_parameter_set_id="6" chroma_format_idc="0" pic_width_in_luma_samples="0"
                    conf_win_left_offset="1" conf_win_right_offset="512" conf_win_top_offset="288" conf_win_bottom_off
                    log2_max_pic_order_cnt_lsb_minus4="4" sps_sub_layer_ordering_info_present_flag="1" log2_min_luma_c
```

Figure 1 Snapshot of an atom xml file of a video track

Following the examples 1 and 2, some more unit tests are created to test most of the cases occurring in the conformance checks implemented in the source code. The complete set of unit tests can be found here [2].

5 References

[1] - WAVE Content Specification, April 2018

[2] - https://github.com/Dash-Industry-Forum/CTAWAVE/tree/master/UnitTests